Reliability Engineering and System Safety 208 (2021) 107406

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

Supervised Dynamic Probabilistic Risk Assessment of Complex Systems, Part 1: General Overview

Tarannom Parhizkar [a,*], Jan Erik Vinnem [a], Ingrid Bouwer Utne [a], Ali Mosleh [b,c]

[a] Department of Marine Technology, Norwegian University of Science and Technology (NTNU), Trondheim, Norway
[b] The B. John Garrick Institute for the Risk Sciences, University of California, Los Angeles, USA
[c] Department of Materials Science & Engineering, University of California, Los Angeles, USA

ARTICLE INFO

Keywords:
Dynamic probabilistic risk assessment
Supervised algorithms
Optimization model
Decision-making process
Emergency situations
Dynamic positioning system

ABSTRACT

Dynamic probabilistic risk assessment (DPRA) is a systematic and comprehensive methodology that has been used and refined over the past decades to evaluate risks associated with complex systems. However, current approaches to construct and execute DPRA models are challenged by high execution time owing to numerous possible scenarios. This issue will affect the execution time of the model, which is in contrast with the aim of modeling. DPRA models must be sufficiently fast to assist decision-making processes in the required time.

In this study, a new method is proposed to enhance the execution times of DPRA models. This method uses optimization algorithms to determine failure scenarios and sort scenarios based on their occurrence probabilities. The most efficient optimization algorithms, considering the nature of the DPRA models, are mixed-integer sequential quadratic programming, modified branch-and-bound algorithm, and modified particle swarm optimization, which are then compared and discussed.

To validate the effectiveness of this method, a simple case study is presented. The results show the effectiveness of the method, which has high accuracy and reduces the execution time significantly (e.g. execution time of risk assessment of 16,464 possible behavior scenarios after an incident in a dynamic positioning system is one fifth of the conventional methods). A detailed supervised DPRA model of dynamic positioning systems and its application on three incidents that occurred in the Norwegian offshore sector in previous years is presented in a subsequent article (Part 2 of 1) (Parhizkar et. al.). Case study results confirm that the supervised DPRA method can be applied to other complex systems so that the dynamic probabilistic risk values can be evaluated quickly and accurately.

1. Introduction

Probabilistic risk assessment (PRA) is a structured method of quantitative risk assessment to navigate the design and operation of systems for achieving a certain safety or operational goal. The National Aeronautics and Space Administration [1], International Atomic Energy Agency [2], US Nuclear Regulatory Commission [3] have provided general guidelines for PRA for managers and practitioners. Recently, many research groups have performed PRA for different applications, including human health [2], airport airside safety [3], and safety in renewable power plants [4]. Mosleh [5] reported the strengths, limitations, and possible improvements of PRA.

The dynamic characteristics and behavior of a system, stochastic processes, operator response times, inspection and testing time intervals, aging of equipment or components, seasonal changes, sequential dependencies of equipment or components, and timings of safety system operations affect system PRAs significantly [6]. Generally, conventional PRA methodologies have limited capacity for analyzing and quantifying the dynamic characteristics of complex systems. Therefore, it is important to develop a method that can address time-dependent effects in PRAs and provide precise estimations. Dynamic PRA (DPRA) has been used to understand unintended time-dependent interactions between system components, including technical, environmental, and organizational factors, over time. Lapp et. al. [7] and Kumamoto et al. [8,9] were among the first pioneers of proposing dynamic probabilistic risk assessment in real case applications. Different methods have been proposed for considering these system interactions in risk assessment [10], of which most are based on the

* Corresponding author: Department of Marine Technology, NTNU, 7491 Trondheim.
E-mail addresses: tarannom.parhizkar@ntnu.no, parhizkar.t@gmail.com (T. Parhizkar).

https://doi.org/10.1016/j.ress.2020.107406
Received 7 February 2020; Received in revised form 25 November 2020; Accepted 11 December 2020
Available online 15 December 2020

dynamic/continuous event tree [11,12,13], dynamic fault tree [14,15], dynamic Bayesian network (BN) methodologies [16,17]. In these methods, instead of a unique event sequence diagram (ESD) connected to specific fault trees (FTs) and BNs, multiple ESDs/FTs/BNs are developed for an incident scenario that can be updated over time according to the environmental and operational characteristics of the system.

Other methods include Monte Carlo [18], Markovian approaches [19], Petri nets [20] and genetic algorithms [21]. Aldemir [22] has performed a review on different types of dynamic probabilistic safety assessment (PSA) methodologies proposed to date.

As presented, DPRA methods are developed by integrating several approaches and are considered as hybrid methods. A hybrid method refers to a method that combines two or more methods for modeling risk scenarios [23]. Zio [24] has studied the concepts and challenges of integrating deterministic and probabilistic safety assessment models. Emerging hybrid methods tend to mix fundamentally different representational and computational techniques. The presented hybrid methods can be utilized to generate operational/failure scenarios of systems over time. A DPRA for a complete nuclear power plant for severe accidents can be performed using a number of computational tools such as [21,25,26,27,28]. These methods could have different level of uncertainty depending on the time constraints for the analysis and available information about the system.

The nature of the complex system behavior is uncertain. As a result, the number of possible failure scenarios after an incident increases significantly. The ability to handle a large number of scenarios without compromising completeness is a challenging problem encountered in simulation-based risk assessment methodologies. This challenge is explained in more detail through a case study in Section 2. Some of the researchers have tried to limit the number of generated/analyzed scenarios utilizing different methods, in [25,26,27,28] investigate clustering machine learning methods and algorithms to analyze the generated large time-dependent data sets of accident scenarios from simulation. In this method, the scenarios with similar behavior are identified. Then each scenario will be associated with a unique cluster.

Nielsen et. al. [29,30] proposed branch and bound optimization method to control the combinatorial state explosion of the DPRA. In 2015, Nielsen et al. [29] applied a Branch-and-Bound optimization algorithm to Dynamic Discrete Event Trees (DDET) in order to address combinatorial explosion. This method utilizes LENDIT (L – Length, E – Energy, N – Number, D –Distribution, I – Information, and T – Time) as well as a set theory to describe system, state, resource, and response (S2R2) sets to create bounding functions for the DET. In this study, a Phenomenological Identification and Ranking Table (PIRT) methodology based on optimization of the DET is discussed. This methodology is used to evaluate modeling parameters important to safety of those failure branches that have a high probability of failure.

In this study, a supervised DPRA method is proposed that considers system dynamics and its dependency upon environmental and organizational factors over time to generate sorted failure scenarios based on occurrence probabilities and/or risk levels, i.e., this method identifies the failure scenarios sequentially based on their probabilities and/or risk significance rather than rank ordering them after identifying the complete set. This capability results in significantly lower computational time.

The proposed method is based on an optimization model, whose generic form is presented in Section 3. Further, the three most suitable solution algorithms for dynamic probabilistic risk models are presented and discussed. In Section 4, a case study is presented where the proposed optimization model is applied to a dynamic positioning system. In Section 5, results are presented, which show that the execution time of the DPRA is reduced significantly in the case study example.

2. Challenges of DPRA methods for complex systems

Complex systems (CSs) are mostly nonlinear and nondeterministic due to the heavily interdependencies among their elements [31]. Considering nonlinearity and stochastic characteristics in risk modeling would enable a more accurate analysis and prediction of the complex system behavior. Such predictions can provide a valuable basis for decision support regarding the need for risk mitigation and accident prevention to operators.

Generally, a complex system behavior is affected by human machine interactions, and operational and environmental conditions [31]. Figure 1 presents an agent-based model of a complex system that includes human behavior, system state, and operational and environmental behavior agents, which are interconnected. The edges present the interconnections and interactions between human-system-environment in real world.

As presented in Figure 1, each agent consists of multiple sub-agents, depending on the modeling approach and abstraction. For instance, the system state, presented in Figure 1, has three sub-agents. The state of the sub-agents is affected by interaction with other sub-agents and is updated over time. The state of sub-agents could take binary or continuous variables. For instance, system state agent could have a sub-agent as “available time”. This sub-agent evaluates the available time before collision happens in a system. In this example, “available time” sub-agent could take binary (low/high) or continuous values.

The type of sub-agent states should be defined based on the scope of the study. The same system with different scopes can result in different types of sub-agents’ states. For instance, if we are interested in the working status of a system, we could have binary states of success or failure. However, if we want to know more about the system degradation level, we could have more states such as partially and/or totally degraded states.

The behavior of a CS is updated over time owing to agents’ and subagents’ interactions. To capture the behavior of a CS over time, system variables should be predicted. System variables include all the states of the sub-agents. In some CSs, sensors and monitoring systems help to capture the state of one or multi sub-agents. For instance, the state of sub-agents presented in “operational and environmental variables” agent could be defined based on the received data from sensors.

The prediction process tries to predict the unknown sub-agent states based on the available information. The prediction process is associated with uncertainty; primarily, the probability distribution of a model prediction is presented as a suitable basis for evaluating the uncertainty of a predicted behavior. To achieve sufficient accuracy, different possible alternative scenarios (possible system behavior scenarios) should be predicted over time. Each possible alternative has a probability of occurrence. To evaluate the risk level of a CS, all the failure

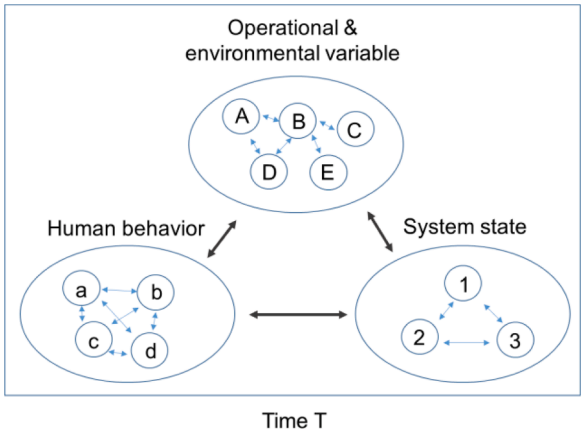

Figure 1. Example of a complex system

modes of these possible alternative scenarios must be considered.

The accuracy and execution time of a DPRA model is highly dependent on the number of predicted possible scenarios. A limited number of predicted scenarios results in an inaccurate model of a system. Increasing the number of predicted scenarios enhances the model accuracy. However, in some cases, it is complicated and sometimes impossible to consider all possible scenarios.

To demonstrate the complexity of the problem, a dynamic positioning (DP) system is presented as an example. The DP system is a computer-controlled system designed for automatically maintaining a vessel or rig's position and heading it using its own propellers and thrusters. A DP system comprises mechanical machineries, sensors, and human interactions. Figure 2 shows the basic components in a DP system.

As shown, a power generation system, a switchboard, and thrusters are connected to the DP control system, which gather data from multiple sensors, including the reference system for position, gyro compass for heading, motion reference units, and environmental sensors (wind). The DP operator (DPO) obtains information from the DP control system using a DP desk, which is a man–machine interface that provides useful information about the components' statuses. Furthermore, the DPO can access other information through communication systems, system alarms, and alarm system traffic light.

The aim of the example presented herein is to model an operator's behavior during emergency situations. The emergency is any incident results in the position loss and/or mechanical damage of the vessel in a short time. Three main decision scenarios exist in an emergency.

1 Keep position: In this scenario, the operator attempts to maintain the vessel position. This task can be performed manually or automatically. In the automatic mode, the DP system maintains the vessel position automatically using a control system and actuators. However, in the manual mode, the operators use a joystick for position and heading control.
2 Disconnection: In this scenario, the DPO cannot maintain the position. Consequently, the DPO attempts to stop the vessel from operating. This task can be performed manually or automatically. In the automatic mode, the DP system is disconnected, and the components are stopped automatically. However, in the manual mode, the system is controlled and disconnected manually, independent of the DP control system.
3 Recovery action: In some situations, sufficient time exists to recover failure components. Generally, after performing the keep position or disconnection, the DP system can be in a safe zone and there would be sufficient time to safely recover faulty components. Only a limited number of recovery actions can be performed during emergency situations. Herein, seven recovery action scenarios are considered.

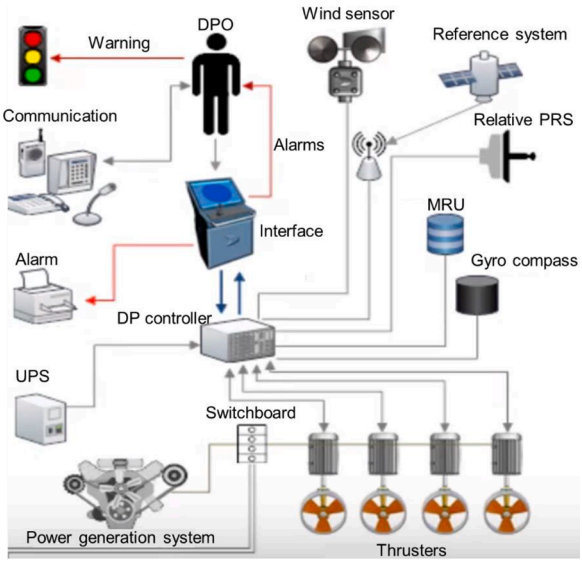

Figure 2. Dynamic positioning system [32]

The agent based conceptual model of the DP system is presented in Figure 3. The DP system has three main agents which are interconnected including "human behavior", "system state", and "operational and environmental variable". The behavior of these agents, and their interconnections with other agents is presented by edges that is governed by complex rules. The whole DP system is modeled by coupling the elements and edges governing principles and rules.

As shown, the "system state" agent has one subagent, which comprises components; the "operational and environmental variable" agent has three subagents, including the position, available time, and alarm; and finally, the "human behavior" agent has 15 subagents, including monitoring, three types of detection (alarm, check position and heading, other visual detections), two types of keep position execution (automatic and manual modes), two types of disconnection execution (automatic and manual modes), and seven recovery action execution (change position reference, recalibrate reference origin, deselect faulty sensor, recover reference system, tune software, start new generator, and restart equipment).

All these sub-agents are governed by simple rules that could be modeled and simulated with acceptable accuracy. Modeling the subagents and their interactions could simulate the agent's behavior. Having agent's behavior and the interactions between agents form the DP complex system behavior and functionality.

In this example, sub-agent models are abstracted and each sub-agent could take two states in the modeling process; for instance, the alarm subagent has on and off states, or the human behavior subagents have "completed" and "in progress" states.

In this example, multiple alternative scenarios could occur after an initial undesired event. In an emergency, by detecting an initial event, the states of operational and environmental variables can be determined according to the sensor data. However, the system and human behavior operation modes can achieve different states. Consequently, the number of possible scenarios after an incident is equal to all combinations of the system and human behavior operation modes, which is equal to 5,376.

$$\begin{aligned} Number\,of\,alternative\,scenarios\,at\,t = &\text{Monitoring states} \times \text{Detection states} \\ &\times \text{Disconnection states} \times \text{Keep position states} \times \text{Recovery states} \\ &\times \text{System states} = 2\times 6\times 4\times 4\times 14\times 2 = 5367 \end{aligned} \tag{1}$$

Where, the number of monitoring, detection, disconnection, keep position, recovery and system states are defined based on the states presented in Figure 3. As time progresses, the number of alternative scenarios increases significantly and can be calculated using Eq. (2).

$$Number\ of\ alternative\ scenarios = (n)^k = 5,376^k \tag{2}$$

The number of scenarios after the k iteration is presented in Table 1.

As shown, the number of scenarios can increase significantly over time. Therefore, agent states should be defined appropriately, as the model size is proportional to the number of agent states. However, not all these combinations are viable, and feasible connections should be defined based on the CS operating characteristics, limitations, and study scope. Even by considering all possible and feasible scenarios, the execution time increases significantly, as numerous possible scenarios should be connected to multiple ETs/ESDs/FTs/BNs/MCs to perform DPRA.

The main challenge is that with the growth of the size of the dynamic systems and the complexity of the interactions between components, it is extremely difficult to enumerate the risky scenarios by traditional ET/FT methods. Simulation methods are often used to solve DPRA problems in complex systems. Most of the simulation models cannot evaluate system

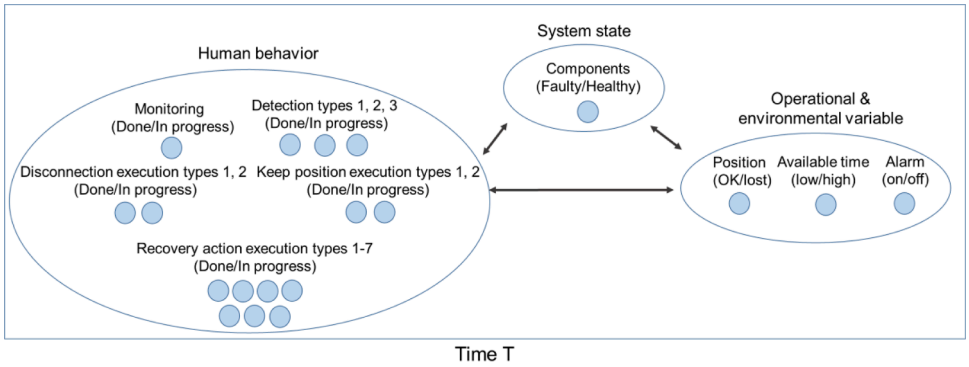

Figure 3. Agents and subagents of a dynamic positioning (DP) system

Table 1
Number of alternative scenarios at each time step

Number of time intervals	1	2	3	4
Number of scenarios	5,376	28e+6	15e+10	83e+13

risk level effectively and they are time-consuming. For instance, it is widely recognized that Monte Carlo simulation is highly inefficient and may result in generating a lot of histories without any information gain [33].

The execution time of DPRA methods are highly important as DPRAs are mostly utilized in emergency situation with a limited timeframe. Thus, a high efficiency simulation engine is often essential to treat realistic systems. Different methods are utilized to reduce the execution time of DPRA methods in complex systems in recent years. However, practical applications to large systems are limited and mostly case dependent. For instance, Lee et. al. [34] proposed a DPRA approach for real-time emergency guidance. This approach is based on a simulation of possible nuclear power plant behavior following an initial event and predict the probability of different levels of off-site release of radionuclides based on deep learning techniques. A large number of scenarios are simulated and used to train the proposed deep learning model.

In our study, an optimization method is proposed to generate more probable scenarios (or scenarios with high risk level). In this method, risky scenarios are analyzed without assessing all possible scenarios. Instead, the desired scenarios are generated, and all the evaluation are performed for the selected scenarios.

Different optimization algorithms could be applied to the optimization model to search over the complex system and find desired scenarios without finding/solving all possible options. As the number of scenarios under study are limited, the execution time would be reduced significantly, and this method could be implemented in emergency situations.

3. Optimal scenario selection

In this study, a new concept is introduced that generates failure scenarios after an incident, considering operation and environmental conditions. In this new approach, all alternative possible scenarios need not be solved; instead, failure scenarios are derived using an optimization model. To sort the failure scenarios without considering all possible scenarios, an optimization concept is introduced. In the optimization concept, an optimization model is developed based on the system's characteristic and constraints to obtain more probable scenarios. In this section, the optimization model and related solution algorithms are presented.

3.1. Optimization model

An optimization problem typically has three main elements. The first is an objective function that is to be maximized or minimized according to the study objective [35]. The objective of this study is to maximize the probability of the scenario occurrence. Maximizing the probability means that the optimization model tries to find a scenario with maximum occurrence probability among all possible scenarios.

$$\text{Maximize } (\text{Occurrence probability}) \tag{3}$$

If we are interested in more than one probable scenario, then the optimization model should be run for several times. At the first run, the model obtains the most probable scenario; subsequently, at the second run, this scenario is saved for later, and the model is executed again to obtain the second most likely scenario. The model is executed until all desired sorted failure scenarios are derived.

Depending on the objective of the study, the objective function could be modified. For instance, the objective of the most DPRA studies is to find scenarios with high risk level, e.g., scenarios with high probability and low consequences and scenarios with low probability and high consequences. In these cases, the objective function is equal to the risk level evaluation. As presented in Eq. (4), in addition to scenario probabilities, scenario consequences should be considered in the objective function.

$$\text{Maximize } (\textit{Occurrence probability} \times \textit{Consequence}) \tag{4}$$

In this case, the model obtains a scenario with the highest risk level; subsequently, this scenario is saved for later, and the model is executed again to obtain the next scenario with the highest risk level. The model is executed until all failure scenarios are sorted based on the risk level.

The second element in an optimization model is decision variables that should be optimized to maximize and/or minimize the objective function. Two types of continuous and discrete decision variables exist [35]. The decision variables in this study are the number and type of events in a failure scenario. These variables are discrete and can be defined as integer variables in the optimization model.

$$x \in \{0, 1\} \tag{5}$$

where 1 and 0 represent the existence and non-existence of a connection or event in an event sequence diagram, respectively.

The third element of an optimization problem is a set of constraints, which are restrictions on the values that the variables can assume [35]. The constraints should be defined according to the system under study. A system dynamic model and all applicable logical rules should be considered. The constraints can be linear or nonlinear functions. Most risk assessment studies have nonlinear constraints. Three mains constraints that can be considered in risk assessment problems are:

Logical constraints:

1 Human cognition rules

Principle-based constraints:

1 Derived from CS simulation model
2 Principle rules of probability and risk assessment

System specification and limitations:

1 Risk criteria
2 Standards that should be followed.

Human cognitive rules include governing rules of human behavior. For instance, decision making process consists of following steps in order: monitoring, detection, diagnosis, decision making and execution. This order implies that execution will be performed after decision making has been performed or decision-making process will be started after diagnosis has been performed. These types of rules should be considered as logical constraints in the optimization process.

Principle based constraints include complex system simulation model and principle rules of probability and risk assessment. This type of constraints presents governing rules of the system. Governing rules could be derived from system physical principle equations and/or data-driven models. These rules represent system behavior and evaluate system performance based on the objective of the study.

The last type of constraints present system specification and limitations. Design and operational limits of a system are fall into this type of constraints, e.g., risk level of the system under operation should always be under a critical value.

Within this broad optimization model, the mathematical properties of risk assessment problems include modeling nonlinear equations considering integer decision variables. These mathematical properties imply that risk assessment optimization problems can be classified as mixed-integer nonlinear programming. It is noteworthy that the optimization of risk assessment problems has not been studied and that the optimization model and the proposed classification are main novelties of this study.

3.2. Optimization algorithms

Mixed-integer nonlinear programming (MINLP) addresses a class of optimization problems with nonlinearities in the constraints and/or the objective as well as integer and/or continuous variables [36]. Multiple solution algorithms have been proposed for MINLP problems [37]. According to the performance of the solution algorithm, the best suitable algorithm was selected for the system under study. In this section, three of the most typical algorithms applicable to risk assessment problems are presented.

3.2.1. Mixed-integer sequential quadratic programming (MISQP)

The sequential quadratic programming (SQP) algorithm is a practical solution algorithm for continuous nonlinear optimization problems. This algorithm converges globally toward an optimal point, and the global minimum can be guaranteed for convex optimization problems [38].

Every SQP method is based on solving a series of continuous quadratic programming (QP) subproblems. At each iteration, a quadratic subproblem is constructed by linearizing the constraints at the current iterate. The objective of the algorithm is to obtain a quadratic approximation of a Lagrangian function that results in an optimal solution [38]. Recent developments in SQP have yielded a modified SQP that can consider integer decision variables in the optimization process, called mixed-integer sequential quadratic programming (MISQP), which can efficiently solve MINLP problems [39].

As SQP algorithm is developed for continuous nonlinear optimization problems, MISQP can solve MINPs with limited number of integer decision variables. In other words, this algorithm is suitable for the risk assessment of systems with a limited number of alternative probable scenarios.

3.2.2. Modified branch-and-bound algorithm

The branch-and-bound algorithm is originally developed for solving mixed-integer linear optimization problems [40]. Branch-and-bound is based on branching, bounding, and fathoming. In the first step, the problem is iteratively divided into a finite number of subproblems. The solution process is based on a series of bounding of the original problem, which is easier to solve. During the bounding process, some branches are decided as fathomed (pruning the branches) [41,40]. A branch is fathomed if its value is less than or equal to p*. p* could be the critical occurrence probability or risk level, which should be defined by experts for each case under study. At each branching points, scenarios with probabilities/risk levels lower than p* would be eliminated.

The algorithm is expanded to solve convex MINLP problems as well. The modified branch-and-bound can consider the nonlinearity of the constraints and/or objective function in the optimization process; consequently, it can solve MINLP problems [42].

Since the branch-and-bound algorithm is developed for solving mixed-integer linear optimization problems, it is more suitable for problems with low nonlinearities. In fact, this algorithm can perform well in the risk assessment of systems with low complexity, i.e., when variables are not highly dependent or when dependencies are negligible.

3.2.3. Modified particle swarm optimization (PSO)

PSO is a population-based optimization algorithm inspired by the social behavior of fish schooling and bird flocking. The PSO algorithm shares many similarities with other evolutionary optimization algorithms, such as genetic algorithms. A main advantage of the PSO algorithm over other evolutionary methods is that only a few parameters need to be adjusted in the PSO algorithm. The algorithm starts with a population of random solutions and searches for the optimal point by updating generations. In other words, the PSO computational method optimizes a problem by iteratively improving a candidate solution by considering a given measure of quality. The candidate solution is affected by the local best-known point, and it is guided toward the best-known positions in the search space, which are updated as better positions are discovered by other particles. This process is expected to move the candidate solution toward the optimal solution [43].

The PSO algorithm is suitable for nonlinear programming with continuous decision variables. However, different algorithms are proposed to modify PSO for handling mixed-integer nonlinear programming. One of the proposed modifications is to adapt random search methods to the integer case by adding a round-off instruction to select the nearest integer of a given real value [44].

Table 2 presents a comparison of solution algorithms based on nonlinearity (objective function and constraints) and decision variable type. In all algorithms, the decision variables can be an integer and/or continuous; however, in the table, the preferred variable that results in a shorter convergence time is presented.

4. Case study

A DP system is considered as an example in this study. The schematics of the agents, subagents, and states of this complex system are presented in Figure 3. The objective function is

Table 2
Comparison between optimization algorithms

	Nonlinearity	Preferred decision variable
Mixed Integer Sequential Quadratic Programming	Low	Continuous
Modified branch-and-bound algorithm	Low	Integer
Modified PSO	High	Continuous

Table 3
Parameters definition

Agent	Sub-agent	State
k=1 Operational & environmental variables	j=1 Position	i=1 Ok/ i=2 Lost
	j=2 Available time	i=1 Low/ i=2 High
	j=3 Alarm	i=1 On/ i=2 Off
k=2 Human behavior	j=1 Monitoring	i=1 Done/ i=2 In progress
	j=2 Detection type 1	i=1 Done/ i=2 In progress
	j=3 Detection type 2	
	j=4 Detection type 3	
	j=5 Disconnection type 1	i=1 Done/ i=2 In progress
	j=6 Disconnection type 2	
	j=7 Keep position type 1	i=1 Done/ i=2 In progress
	j=8 Keep position type 2	
	j=9 Recovery action type 1	i=1 Done/ i=2 In progress
	j=10 Recovery action type 2	
	j=11 Recovery action type 3	
	j=12 Recovery action type 4	
	j=13 Recovery action type 5	
	j=14 Recovery action type 6	
	j=15 Recovery action type 7	
k=3 System state	j=1 Components	i=1 Faulty/ i=2 Healthy

$$Max\sum_{k=1}^{l}\sum_{j=1}^{m}\sum_{i=1}^{n}\left(p_{ijk}\times x_{ijk}\right)+\sum_{k=1}^{l}\sum_{j=1}^{m}\sum_{i=1}^{n}\left(p_{ijk-ijk}\times x_{ijk-ijk}\right)_{t}+\sum_{k=1}^{l}\sum_{j=1}^{m}\sum_{i=1}^{n}\left(p_{ijk-ijk}\times x_{ijk-ijk}\right)_{t-(t+1)}, \quad (6)$$

Where the first term presents the occurrence probability of each subagents, the second term presents the connection probability between subagents at time t, and the last term presents connection probabilities between subagents at times t and t + 1.

- p_{ijk} presents the probability of state i for subagent j and agent k.
- x_{ijk} is a binary variable {1,0} that indicates the existence and/or non-existence of state i for subagent j and agent k.
- n is the number of states.
- m is the number of subagents.
- l is the number of agents.
- $(p_{ijk-ijk})_t$ presents the connection probability of state i, subagent j, agent k with state i, subagent j, agent k at time t.
- $(x_{ijk-ijk})_t$ is a binary variable {1,0} that indicates the existence and/or non-existence of the connection between state i, subagent j, agent k and state i, subagent j, agent k at time t.
- $(p_{ijk-ijk})_{t-(t+1)}$, presents the connection probability of state i, subagent j, agent k with state i, subagent j, agent k over time t to t+1.
- $(x_{ijk-ijk})_{t-(t+1)}$ is a binary variable {1,0} that indicates the existence and/or non-existence of the connection between state i, subagent j, agent k and state i, subagent j, agent k over time t to t+1.

Table 3 presents the i, j, and k of each agent, subagent, and state for the presented case study.

The first constraint of the case study is that only one active (x = 1) state exists at a time. For instance, an alarm on and off signal cannot occur simultaneously. This constraint is presented in Eq. (7).

Table 4
Agent dependency rules at time t

State of agent at time t	Activated agent at time t	Connection probability
Alarm (Off)	Keep position (In progress)	$(P_{231\text{-}242})_t$
Alarm (On)	Detection (In progress)	$(P_{131\text{-}222})_t$
Available time (High)	Recovery action execution (In progress)	$(P_{221\text{-}252})_t$
Components (Faulty)		$(P_{113\text{-}252})_t$
Available time (Low)	Disconnection execution (In progress)	$(P_{121\text{-}232})_t$
Components (Faulty)		$(P_{113\text{-}232})_t$
Alarm (On)	Keep position (In progress)	$(P_{131\text{-}242})_t$
Components (Healthy)		$(P_{213\text{-}242})_t$
Position (Lost)	Keep position (In progress)	$(P_{211\text{-}242})_t$
Components (Healthy)		$(P_{213\text{-}242})_t$
Position (Lost)	Disconnection execution (In progress)	$(P_{211\text{-}232})_t$
Components (Faulty)		$(P_{113\text{-}232})_t$
Keep position (In progress)	Recovery (In progress)	$(P_{242\text{-}252})_t$

$$\sum_{i=1}^{n}\left(x_{ijk}\right)=1\ ,\ for\ j=1,2,\ ..,\ m;\ k=1,\ 2,\ ..,\ l \quad (7)$$

The other constraints including human cognition rules and principle-based constraints present the connection probabilities between agent states at time t or over time t to t+1. According to the logic of the CS, the possible connections between agent states can be defined by experts. The connection between agents at time t for the case under study is presented in Table 4. These values are considered in the second term of the objective function, presented in Eq. (6).

The dependency rules presented in Table 4 convey the probabilities, presented as connection probabilities. These probabilities depend on the agent characteristics at time t and could be quantified based on the expert judgment and/or Bayesian networks. In this study, it is assumed all the connection probabilities are equal to 0.95.

In addition to connections at time t, agents can be connected to other agents at time t + 1. For instance, if the detection is completed, keep position, disconnection, or recovery actions should be performed. The dependency rules over time are presented in Table 5. These probabilities are applied to the third term of the objective function, presented in Eq. (6).

The probabilities of events and connections are a function of the operational and environmental conditions in question. Based on the ESD, FTs, BNs and simulation model of the system under study, the probabilities are quantified. In this example, the probability of events (sub-agents work properly) is assumed to be equal to 0.95. The quantification process of the events probability and connection probabilities are elaborated for a DP complex system in the subsequent article (Part 2) [45].

In addition, the state of some agents can be defined based on sensor data and operating conditions. An example of operating conditions over four time interval is presented in Figure 4. Generally, these values are sensor information from a DP system and time intervals presents the intervals that we get data from sensors.

The operating conditions presented in Figure 4 are x values related to the subagents at each time interval. For instance, at the first time interval:

$$x_{131}=1(\text{Alarm is On}) \quad (8)$$

Table 5
Agent dependency rules over time

State of agent at time t	Activated agent at time t+1	Connection probability
Detection (Done)	Execution (keep position, disconnection, recovery actions)	$(P_{122\text{-}2j2})_{t\text{-}(t+1)}$
Execution (Done)	Monitoring	$(P_{1j2\text{-}212})_{t\text{-}(t+1)}$

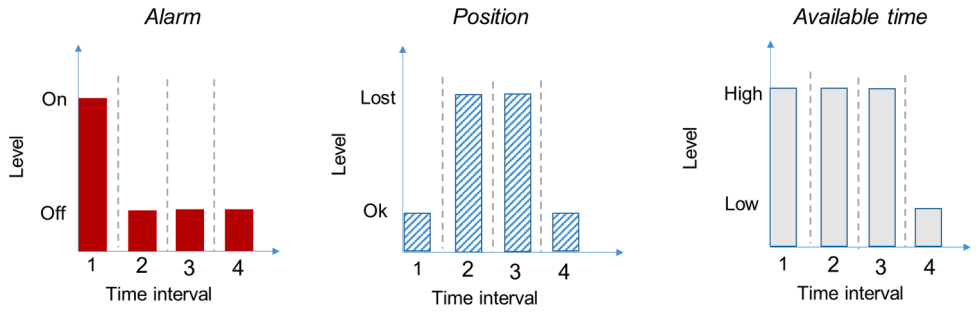

Figure 4. Operating conditions over time for the example

Table 6
Possible scenarios at each time interval

Time interval 1	Time interval 2	Time interval 3	Time interval 4
Detection[a] (3)	Keep position[b] (2) Detection (3) Disconnection[b] (2)	Keep position (2) Detection (3) Recovery[c] (7) Disconnection (2)	Keep position (2)

[a] : Three types including alarm, check position and heading, and other visual detections.
[b] : Two types including automatic and manual.
[c] : Seven types including change position reference, recalibrate reference origin, deselect faulty sensor, recover reference system, tune software, start new generator, and restart equipment.

$$x_{111} = 1(\text{Position is OK}) \tag{9}$$

$$x_{221} = 1(\text{Available time is High}) \tag{10}$$

According to Eq. (7):

$$x_{231} = 0(\text{Alarm is Off}) \tag{11}$$

$$x_{211} = 0(\text{Position is Lost}) \tag{12}$$

$$x_{121} = 0(\text{Available time is Low}) \tag{13}$$

At each time interval, x and p are generated according to Table 4, Table 5, and Figure 4. As shown, at the first level, the alarm is on, the position is "OK", and the time is sufficient. According to Table 4, at this time interval, detection would be activated. At the next time step, the outcomes of the first time interval are used as inputs; in addition, according to the dynamic simulation, the alarm is off, the position is "lost," and the available time is high. According to this information, the possible activated subagents are generated accordingly. All possible subagents for all the time intervals are presented in Table 6. The number of alternative scenarios of each subagent (presented in Figure 3) is shown in parentheses.

According to Table 6, the numbers of possible scenarios corresponding to each time interval are 3, 7, 14, and 2. At each time interval, one scenario could exist. Therefore, the number of all combinations of these scenarios in this duration (four time interval) is the multiple of the number of possible scenarios at each time interval which is equal to 588.

$$\textit{Number of possible scenarios after 4 time intervals} = \prod_{i=1}^{4} \text{Number of scenarios at time interval}\, i = 3 \times 7 \times 14 \times 2 = 588 \tag{14}$$

As time progresses, the number of scenarios increases significantly. In all the conventional methods, the probability of all the generated scenarios should be evaluated, which could result in a long execution time. However, in the proposed concept, an optimization algorithm is applied to the model.

In the optimization process, optimal x values for all subagents are generated to obtain the most probable failure scenario. The optimal x values are binary values that determine the existence and nonexistence of the subagent states of the most probable scenario. After defining the most probable scenario, the optimization model is executed again to obtain the second most probable scenario and corresponding x values are generated. The process is repeated until all failure scenarios are generated. It is noteworthy that in this method, failure scenarios are generated and sorted without solving all possible scenarios. Consequently, the "optimal" scenarios, the most likely scenarios, can be generated in a short execution time.

Figure 5 presents the modified PSO optimization algorithm for the case under study. First, PSO parameters are initialized and random position and velocity are selected for each particle. Then, the fitness function for each particle is evaluated. The fitness function is the presented objective function in Eq. (6). Based on the obtained values, the velocity and position of each particle is evaluated. Optimal position and velocity at the time iteration t, is the local optimal point that guides the next iteration toward global optimal solution in the study timeline. This iterative process is repeated until the stopping criteria are met. The derived optimal x values are recorded as optimal values of interval t (mission duration).

5. Results and discussion

To evaluate the effectiveness of the proposed optimization method, an example is presented based on the case study discussed in Section 4. The connection probabilities provided in Tables 4 and 5 are considered as a function of system and environmental characteristics (Figure 4). In section 5.1, more probable failure scenarios after different time intervals are presented, and performance of the proposed method is compared with conventional methods. In Section 5.2 a sensitivity analysis on DP system characteristics is performed and presented.

5.1. Performance evaluation

In this section, the proposed optimization method is utilized to generate failure scenarios related to operating and environmental conditions, presented in Figure 4. The modified PSO algorithm is used as the solution algorithm, and the failure scenarios are sorted based on their occurrence probability. According to Table 6 and Figure 4, the numbers of possible scenarios corresponding to each time interval are 3, 7, 14, and 2.

The total number of alternative scenarios after two time interval is equal to the combination of 3 and 7 states, which is 21 scenarios. In order to find more probable scenarios, modified PSO algorithm is applied to the objective function, and optimal x values are derived with considering all constraints (equations and assumptions presented in Section 4). X values could take 0 and 1; values equal to one indicate active states at each time interval that result in higher failure

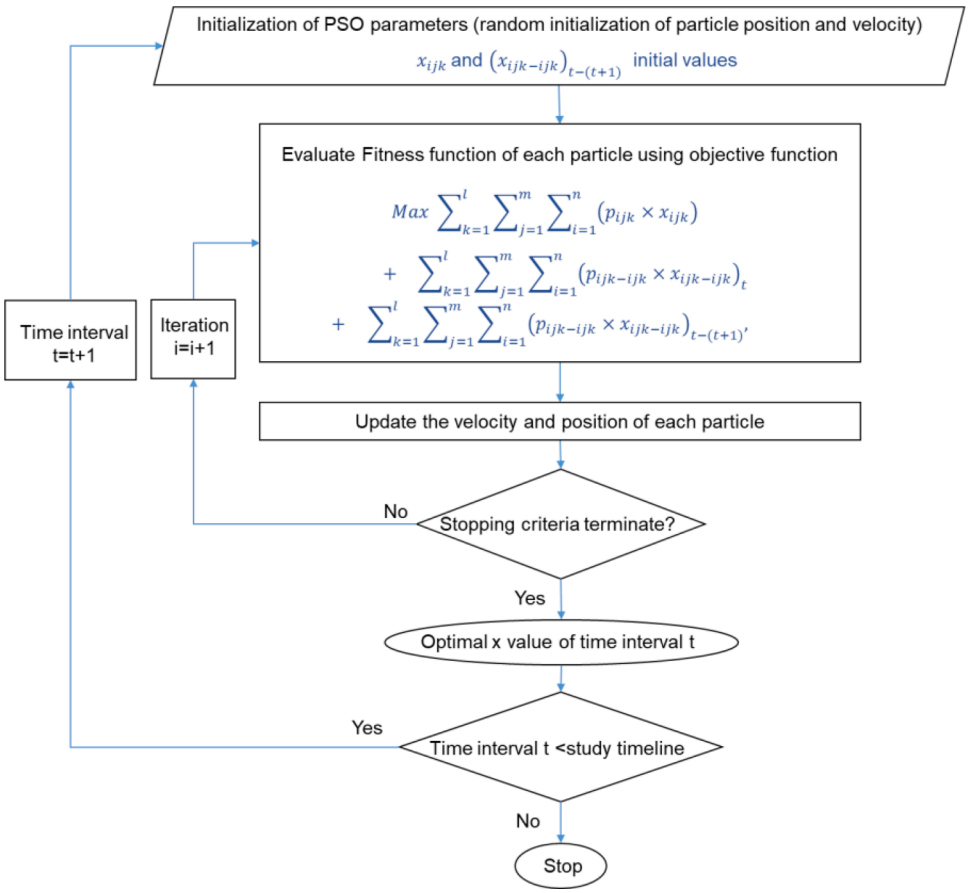

Figure 5. the modified PSO optimization algorithm

Table 7
Failure scenarios after two time intervals

No.	Time interval 1	Time interval 2
1	Position checked properly	Failure of manual keep position
2	Alarm detected	Failure of manual keep position
3	Position checked properly	Failure of auto keep position
4	Alarm detected	Failure of auto keep position
5	Visual detection done perfectly	Failure of manual keep position
6	Visual detection done perfectly	Failure of auto keep position

probabilities.

Table 7 presents the optimal active scenarios after two time intervals. Time intervals are predefined[1] and presented in Figure 4. At each time interval, environmental and operational conditions of the system are updated according to Figure 4; and, most probable scenarios (active states of each sub-agent) will be derived using the optimization algorithm. As the objective function is maximizing failure probability, the derived scenarios are sorted based on the failure probabilities.

The six presented scenarios are sorted failure scenarios in the two time intervals. After the initiating event, the operator is informed regarding the situation in three different methods (alarm detection, position check, and/or other visual detection methods), which are presented in Figure 3. The information from this level is used in the next level, which is execution (Table 5). As mentioned in Section 1, the keep position execution in DP systems comprises two modes: automatic and manual. All combinations of these scenarios in time intervals 1 and 2 generate six scenarios, as presented in Table 7.

[1] Time intervals present the intervals that we get data from sensors

As shown, the most probable failure scenario (number 1) is that after the initiating event, the operator has checked the position and heading properly; however, in the next time interval, the manual keep position is failed. In the least likely failure scenario (number 6), the operator performs visual detections well; however, at the next time interval, the automatic keep position is failed.

As time progresses, the number of scenarios (possible system behaviors) increases (Table 6). The total number of alternative scenarios after three time interval is equal to the combination of 3, 7, 14 states, which is 294 scenarios; and the total alternative scenarios after four

Table 8
Failure scenarios after three time intervals

No.	Time interval 1	Time interval 2	Time interval 3
1	Alarm detected	Auto DC	Failure of change position reference
2	Alarm detected	Auto DC	Failure of recalibrate reference origin
3	Alarm detected	Auto DC	Failure of deselect faulty sensor
4	Alarm detected	Auto DC	Failure of reference system recovery
5	Alarm detected	Auto keep position	Failure of change position reference
6	Alarm detected	Auto keep position	Failure of recalibrate reference origin
7	Alarm detected	Auto keep position	Failure of deselect faulty sensor
8	Alarm detected	Auto keep position	Failure of reference system recovery
9	Position checked properly	Auto DC	Failure of change position reference
10	Position checked properly	Auto DC	Failure of recalibrate reference origin

Table 9
Failure scenarios after four time intervals

No.	Time interval 1	Time interval 2	Time interval 3	Time interval 4
1	Alarm detected	Auto DC	Change position reference	Failure of manual kp*
2	Alarm detected	Auto DC	Recalibrate reference origin	Failure of manual kp*
3	Alarm detected	Auto DC	Deselect faulty sensor	Failure of manual kp*
4	Alarm detected	Auto DC	Reference system recovery	Failure of manual kp*
5	Alarm detected	Auto kp*	Change position reference	Failure of manual kp*
6	Alarm detected	Auto kp*	Recalibrate reference origin	Failure of manual kp*
7	Alarm detected	Auto kp*	Deselect faulty sensor	Failure of manual kp*
8	Alarm detected	Auto kp*	Reference system recovery	Failure of manual kp*
9	Position checked properly	Auto DC	Change position reference	Failure of manual kp*
10	Position checked properly	Auto DC	Recalibrate reference origin	Failure of manual kp*

* : kp refers to keep position

interval is 588 (combination of 3, 7, 14, and 2 states at each time interval). Tables 8 and 9 present more probable failure scenarios after three and four time intervals, respectively.

It should be noted that the more probable scenarios after two (Table 7), three (Table 8), and four (Table 9) time intervals are mutually exclusive. The optimization model takes the time span as an input and according to the corresponding operating and environmental conditions (Figure 4) generates more probable failure scenarios with the same duration as the input time span. For instance, in a time span consisting of two intervals, the optimal results present more probable failure scenarios that last two time intervals. This could be seen in the last column of the above mentioned tables which presents a failure state.

As presented in Table 8, in more probable failure scenarios, an alarm is detected; subsequently, a vessel is disconnected automatically. After the disconnection, different recovery actions are performed and failed. In scenarios 5–8, an alarm is detected; subsequently, automatic keep position is performed. Next, failure in recovery actions occurs. It is assumed that the incident and alarm are related to the reference system; hence, as shown, most of the failure modes are related to the reference system.

In order to show the effectiveness of the proposed methodology, a comparison between execution time of a conventional DPRA (dynamic ESD) and the supervised (optimization based) DPRA methods is performed. These two methods are applied to the presented case study in section 4, and execution times at different time intervals are recorded. The operating and environmental conditions of both methods are considered to be the same.

In the conventional DPRA method, all possible scenarios are generated at each time interval. The probability of each scenario is calculated using the dynamic ESD/FT/BN of the DP system [46,47], and the scenarios are sorted based on their occurrence probability. In the supervised DPRA method, more probable scenarios are generated using the modified PSO based optimization algorithm, then failure probabilities are evaluated using the same (as conventional method) dynamic ESD/FT/BN of the DP system [46,47]. In these cases, the cut-off value is set to 0.001; and ten more probable scenarios are generated as the probabilities of the two last scenarios (9 and 10), in all cases, are less than 0.001.

Table 10 presents the execution time of these two methods for different time intervals (K). As can be seen, the execution time of the optimization-based method (supervised DPRA) is almost the same for all time intervals. In addition, as presented, at lower time intervals, the execution time of the conventional methods is lower. That is primarily because of the lower number of possible scenarios in these time intervals. As the number of possible scenarios increases, the execution time of the conventional method (dynamic ESD) increases significantly. However, the execution time of the optimization method increases slightly. In this example, the modified PSO is used as the solution algorithm. However, it should be mentioned that other solution algorithms would result in almost the same execution time owing to the simplicity of the presented example. In more complex systems, an efficient solution algorithm should be selected based on the model characteristics and algorithm features. Table 2 presents a guideline for selecting the solution algorithm based on the system nonlinearity level and decision variable type.

5.2. Sensitivity analysis

Sensitivity analysis can identify the dependency of outputs on a particular input. In other words, sensitivity analysis determines the effect of the fluctuation of an independent variable (model input) on a particular dependent variable (model output) under a given set of assumptions. Sensitivity analysis can be performed to explore the robustness and accuracy of a new methodology. Hence, one of the model parameters was changed assuming all the other parameters remained constant; the method was executed, and the changes in outputs were analyzed [48].

In this section, it is assumed that the DP system exhibits technical problem (initiating event), and a component has failed. This assumption affects ESD, FTs and BNs of the system and it will update events and connections probabilities. As mentioned before, in Section 5.1, the results are derived based on the assumption that sub-agents including "components" work properly with the probability of 0.95. In this section, it is assumed that the probability of "components" sub-agent works properly is 0.10, i.e., "components" sub-agent fails with the probability of 0.90.

With this assumption, the PSO based optimization model is executed and the results (more probable scenarios) for two, three, and four time

Table 11
Failure scenarios after two time intervals

No.	Time interval 1	Time interval 2
1	Alarm detected	Failure of auto keep position
2	Alarm detected	Failure of manual keep position
3	position and heading checked properly	Failure of auto keep position
4	position and heading checked properly	Failure of manual keep position
5	visual detection done perfectly	Failure of auto keep position
6	visual detection done perfectly	Failure of manual keep position

Table 10
The number of possible alternative scenarios and execution time at each time interval

Number of intervals		K=1	K=2	K=3	K=4	K=5[a]	K=6[b]
Number of possible scenarios		3	21	294	588	8232	16,464
Execution time	Conventional method	<10s	<10s	~10s	~1min	~2min	~5min
	Optimization based method	10s<t<1min					

[a] : It is assumed that after interval four, the interval three is repeated. The number of alternative scenarios is equal to the combination of 3, 7, 14, 2, and 14.
[b] : It is assumed that after interval four, interval three and four are repeated. The number of alternative scenarios is equal to the combination of 3, 7, 14, 2, 14, and 2.

Table 12
Failure scenarios after three time intervals

No.	Time interval 1	Time interval 2	Time interval 3
1	Alarm detected	Manual DC	Failure of change position reference
2	Alarm detected	Manual DC	Failure of deselect faulty sensor
3	Alarm detected	Manual DC	Failure of reference system recovery
4	Alarm detected	Manual keep position	Failure of change position reference
5	Alarm detected	Manual keep position	Failure of deselect faulty sensor
6	Alarm detected	Manual keep position	Failure of reference system recovery
7	Position checked properly	Manual DC	Failure of change position reference
8	Position checked properly	Manual DC	Failure of deselect faulty sensor
9	Position checked properly	Manual DC	Failure of reference system recovery
10	Position checked properly	Manual keep position	Failure of change position reference

Table 13
Failure scenarios after four time intervals

No.	Time interval 1	Time interval 2	Time interval 3	Time interval 4
1	Alarm detected	Manual DC	Recalibrate reference origin	Failure of auto kp*
2	Alarm detected	Manual DC	Start new generator	Failure of auto kp*
3	Alarm detected	Manual kp*	Recalibrate reference origin	Failure of auto kp*
4	Alarm detected	Manual kp*	Start new generator	Failure of auto kp*
5	Alarm detected	Manual DC	Recalibrate reference origin	Failure of manual kp*
6	Alarm detected	Manual DC	Start new generator	Failure of manual kp*
7	Alarm detected	Manual kp*	Recalibrate reference origin	Failure of manual kp*
8	Alarm detected	Manual kp*	Start new generator	Failure of manual kp*
9	Position checked properly	Manual DC	Recalibrate reference origin	Failure of auto kp*
10	Position checked properly	Manual DC	Start new generator	Failure of auto kp*

* : kp refers to keep position

intervals are presented in Tables 11, 12, and 13, respectively. The optimal results after two, three and four time intervals present more probable failure scenarios that last two, three and four time intervals, respectively.

As shown in Table 7, the failure of the manual keep position has a higher occurrence probability compared with the automatic keep position. However, in the presented example in this section with higher component failure rates, the failure of automatic keep position has a higher occurrence probability compared with the manual mode, as presented in Table 11. This is because in the automatic keep position, more technical components are involved compared with the manual mode. As the components have higher failure probability, the automatic keep position failure rate increases. Consequently, the automatic keep position has a higher occurrence probability than manual mode.

Table 12 presents the failure scenarios after three time intervals. When comparing this table with Table 8, it is clear that most of the successful disconnections and keep positions presented in Table 12 are performed manually. However, in Table 8, successful automatic mode scenarios are more frequent owing to the higher reliability of the components.

The failure scenarios after four time intervals are presented in Table 13. When comparing this table with Table 9, it is clear that manual modes are more frequent at time interval 2, and the automatic modes are more frequent at time interval 4. As mentioned before, time interval 2 presents successful scenarios; therefore, the manual modes have a higher success probability compared with the automatic modes owing to high technical failure rates. Meanwhile, time interval 4 presents the failure scenarios; as shown, automatic modes are more likely to occur, as presented in Table 13.

6. Discussion

6.1. Model applicability domain

In this study, an optimization method to assist DPRA models is introduced. The proposed method is applicable to DPRA modeling methods, such as the dynamic fault tree, dynamic Bayesian belief network, dynamic event sequence diagram, and hybrid methods (Section 1). The optimization algorithm searches among the possible operation scenarios of a complex system and finds the more desirable ones. Depending on the scope of the study, the objective function to generate desired scenarios is defined. The objective could be set to find the more probable scenarios or scenarios with high consequences (Section 3.1.).

After defining the objective function, the optimization algorithm search over all possible scenarios and find more desired ones, without evaluating the risk level of all feasible scenarios. Then, the desired scenarios could be further analyzed using any probabilistic risk assessment methods (Section 4).

The optimization algorithm helps to reduce the execution time of any DPRA models by reducing the number of scenarios under study (Section 5.1.). Hence, it is highly valuable for risk-informed decision making in emergencies and situations with time restrictions.

6.2. Optimization model extension

6.2.1. Objective function

The proposed optimization method can generate and sort failure scenarios without solving all possible alternative scenarios after an incident occurs. In this study, the objective function of the optimization model defines the more likely scenario. However, the objective function can be updated according to the goal of the study, such as obtaining the least probable scenarios or scenarios with high risk levels (considering consequences as well). In these cases, the methodology remains the same, but the objective function should be updated according to the goal of the system (Section 3.1.). For instance, to obtain high-risk scenarios, the objective function is

$$\text{Maximize } (\text{Occurrenceprobability} \times \textit{Consequence}) \tag{15}$$

6.2.2. Constraints

All the principle rules, constraints, and limitations in the system should be considered as optimization model constraints. As the number of constraints increases, the feasible region of possible scenarios would be restricted. Meanwhile, increasing the number of constraints can increase the nonlinearity of the problem, resulting in multiple local (not global) optimal solutions, which necessitate using evolutionary optimization algorithms. Constraints affect the solution algorithm and execution time significantly. Hence, experts should define and simplify system constraints to decrease model nonlinearity while reflecting the complex system behavior with acceptable accuracy (Section 3.1).

6.2.3. Decision variable

The decision variable of the optimization model is the probable failure scenario. This variable is an integer value, e.g., existence and/or non-existence of an event in a scenario. Therefore, mixed-integer nonlinear programming is proposed as a practical solution algorithm

(Section 3.1).

6.2.4. Solution algorithm

The solution algorithm depends on equation linearity and decision variable type. As most risk assessments are performed on complex systems, which include high levels of nonlinearity, evolutionary algorithms are the best methods. In this study, three main evolutionary algorithms that can handle nonlinear integer decision variables are proposed. These three algorithms are presented as examples of applicable optimization algorithms in this field. However, other evolutionary algorithms that converge to a global optimal solution faster may exist, depending on the complexity of system behavior (Section 3.2).

6.3. Future of DPRA models

The future of DPRA is presented in [49]. A main challenge is related to the development of the DPRA in a complex system, as outlined in Section 2. Risk assessment models are well stablished for cases with considerable data and system behavior information. In this regard, multiple statistical and probabilistic tools can be used to provide valuable information for decision support in many types of application [47].

However, a risk-informed decision-making process is primarily regarding situations characterized by large uncertainties. These situations necessitate the assessment of multiple alternative possible scenarios and, occasionally, unknown scenarios (Table 1). It is a key challenge for the risk field to develop frameworks for this purpose.

DPRA has garnered increasing attention in recent years. Chang et al. [50] performed an in-depth study on this concept, in particular the effect of human factor uncertainty over time in risk assessment. According to their study, possible alternative scenarios increase with uncertainties and time. In the current study, an optimization concept was introduced to generate the most and/or least probable scenarios without considering all alternative scenarios. This concept decreases the execution time of risk assessment significantly. Consequently, risk-informed decision making can be performed in a short time and in emergency situations, with the required acceptable accuracy.

The proposed concept is applicable to systems under uncertainty. However, risk-informed decision-making is, to an increasing extent, regarding unknown situations, such as incidents in a self-learning complex system. Consequences and recovery plans in these systems change over time. In these situations, system behavior and principle rules are updated continuously. We must further develop PRA models that can capture changes linked to the system historical data and governing principles. The available probabilistic dynamic risk approaches and methods would not be feasible in this regard [33]. Risk assessment methods need not be extended in a predictive manner, including attention to system behavior over time. Prediction methods and analysis can be considered as new approaches for improving DPRA methods.

7. Conclusion

A new method utilizing optimization algorithms to generate and sort the failure scenarios of a system after an incident is presented herein. The main challenges of DPRA methods are extensive execution time owing to system complexity, component interdependency, and dynamic behavior of the system. Hence, the new method was proposed, and three optimization algorithms were suggested based on the nature of the DPRA models, which are based on mixed-integer nonlinear programming methods.

The method is applied to a simple case study, and results indicate the usefulness of the method, as it reduced the execution time significantly. Furthermore, the results showed as the time period increased, the difference in execution time between the conventional and proposed methods increased significantly, e.g., the execution time of the proposed method was one-fifth that of the conventional methods after six time intervals, with 16,464 possible scenarios.

Moreover, the method is utilized to model a dynamic positioning complex system, presented in the subsequent article (Part 2) [45], and the developed model is applied to three incidents that occurred in the Norwegian offshore sector. The results, presented in part 2, show that the model can predict the most probable scenarios with an acceptable accuracy in a very short time.

This added capability to the DPRA methods would enable a more complete understanding of incidents and their probable consequences. A successful implementation of the method would enable the method to simulate and obtain the risk level of complex systems accurately in a short execution time. Furthermore, using this method, operators could monitor the risk level of all possible failure scenarios in real time as well as make better decisions in emergency situations.

Credit author statement

Tarannom Parhizkar: Conceptualization, Formal analysis, Validation, Data curation, Modeling, Writing- Original draft preparation. **Jan Erik Vinnem:** Conceptualization, Writing- Review and Editing, Supervision. **Ingrid Bouwer Utne:** Conceptualization, Writing- Review and Editing, Supervision. **Ali Mosleh:** Conceptualization, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Stamatelatos M, Dezfuli H. Probabilistic Risk Assessment Procedures Guide for NASA Managers and Practitioners. Washington, DC: NASA; 2011.
[2] Barrio-Parra F, Izquierdo-Díaz M, Dominguez-Castillo A, Medina R, De Miguel E. Human-health Probabilistic Risk Assessment: The Role of Exposure Factors in An Urban Garden Scenario. Landscape and Urban Planning 2019;185:191–9.
[3] Ketabdari M, Giustozzi F, Crispino M. Sensitivity Analysis of Influencing Factors in Probabilistic Risk Assessment for Airports. Safety Science 2018;107:173–87.
[4] Habib A, Sou C, Hafeez HMuhammad, Arshad A. Evaluation of The Effect of High Penetration of Renewable Energy Sources (RES) on System Frequency Regulation using Stochastic Risk Assessment Technique (an approach based on improved cumulant). Renewable Energy 2018;127:204–12.
[5] Mosleh A. PRA: A Perspective on Strengths, Current Limitations, and Possible Improvements. Nuclear Engineering and Technology 2014;46(nr. 1):1–10.
[6] Chang YHJ, Mosleh A. Cognitive Modeling and Dynamic Probabilistic Simulation of Operating Crew Response to Complex System Accidents. Part 2: IDAC Performance Influencing Factors Model. Reliability Engineering & System Safety 2007;92(nr. 8):1014–40.
[7] Lapp SA, Powers GJ. Computer-aided Synthesis of Fault-Trees. IEEE Transactions on Reliability 1977;26(1):2–13.
[8] Kumamoto H, Henley EJ. Top-down Algorithm for Obtaining Prime Implicant Sets of Non-Coherent Fault Trees. IEEE Transactions on reliability 1978;27(4):242–9.
[9] Kumamoto H, Henley EJ. Safety and Reliability Synthesis of Systems with Control Loops. AIChE Journal 1979;25(1):108–13.
[10] Villa V, Paltrinieri N, Khan F, Cozzani V. Towards Dynamic Risk Analysis: A Review of The Risk Assessment Approach and Its Limitations In The Chemical Process Industry. Safety Science 2016;89:77–93.
[11] Devooght J, Smidts C. Probabilistic Reactor Dynamics—I: The Theory of Continuous Event Trees. Nuclear science and engineering 1992;111(3):229–40.
[12] Zhou J, Reniers G, Khakzad N. Application of Event Sequence Diagram to Evaluate Emergency Response Actions During Fire-Induced Domino Effects. Reliability Engineering & System Safety 2016;150:202–9.
[13] Swaminathan S, Smidts C. The Event Sequence Diagram Framework for Dynamic Probabilistic Risk Assessment. Reliability Engineering & System Safety 1999;63(nr. 1):73–90.
[14] Vesely WE, Goldberg FF, Roberts NH, Haasl DF. Fault Tree Handbook (No. NUREG-0492). Washington DC: Nuclear Regulatory Commission; 1981.
[15] Gascard E, Simeu-Abazi Z. Quantitative Analysis of Dynamic Fault Trees by Means of Monte Carlo Simulations: Event-Driven Simulation Approach. Reliability Engineering & System Safety 2018;180:487–504.
[16] Lei Y, Lee J. Bayesian Belief Network-based Approach for Diagnostics and Prognostics of Semiconductor Manufacturing Systems. Robotics and Computer-Integrated Manufacturing 2012;28(nr. 1):66–74.
[17] Sterritt R, Marshall AH, Shapcott CM, McClean SI. Exploring Dynamic Bayesian Belief Networks for Intelligent Fault Management Systems, i. In: Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics.'cybernetics evolving to systems, humans, organizations, and their complex interactions' 5. IEEE; 2000. p. 3646–52. *cat. no. 02*000.

[18] Rabiti C, Mandelli D, Cogliati J, Kinoshita R. Mathematical Framework for The Analysis of Dynamc Stochastic Systems With The Raven Code (No. INL/CON-13-28225). Idaho National Laboratory (INL); 2013.
[19] Aldemir T. Computer-Assisted Markov Failure Modeling of Process Control Systems. IEEE Transactions on reliability 1987;36(1):133–44.
[20] Németh E, Bartha T, Fazekas C, Hangos KM. Verification of A Primary-to-Secondary Leaking Safety Procedure in A Nuclear Power Plant using Coloured Petri Nets. Reliability Engineering & System Safety 2009;94(5):942–53.
[21] Vorobyev Y, Kudinov P. Development and Application of A Genetic Algorithm Based Dynamic PRA Methodology to Plant Vulnerability Search. In: International Topical Meeting on Probabilistic Safety Assessment and Analysis. 1; 2011. p. 559–73.
[22] Aldemir T. A Survey of Dynamic Methodologies for Probabilistic Safety Assessment of Nuclear Power Plants. Annals of Nuclear Energy 2013;52:113–24.
[23] Madden BP. The Hybrid Model for Concept Development: Its Value for The Study of Therapeutic Alliance. Advances in Nursing Science. 1990.
[24] Zio E. Integrated Deterministic and Probabilistic Safety Assessment: Concepts, Challenges, Research Directions. Nuclear Engineering and Design 2014;280:413–9.
[25] Mandelli D, Maljovec D, Alfonsi A, Parisi C, Talbot P, Cogliati J, Rabiti C. Mining Data in a Dynamic PRA Framework. Progress in Nuclear Energy 2018;108:99–110.
[26] Maljovec D, Wang B, Pascucci V, Bremer PT, Mandelli D. Analyzing Dynamic Probabilistic Risk Assessment Data Through Topology-Based Clustering. In: ANS PSA 2013 International Topical Meeting on Probabilistic Safety Assessment and Analysis; 2013.
[27] Di Maio F, Stasi M, Zio E, Mandelli D, Aldemir T. Identification of Faults in A Level Control Dynamic System. In: ANS NPIC HMIT 2009 Topical Meeting-Nuclear Plant Instrumentation, Controls, and Human Machine Interface Technology; 2009. p. 1228–39.
[28] Maljovec D, Liu S, Wang B, Mandelli D, Bremer PT, Pascucci V, Smith C. Analyzing Simulation-based PRA Data Through Traditional and Topological Clustering: A BWR Station Blackout Case Study. Reliability Engineering & System Safety 2016;145:262–76.
[29] Nielsen J, Tokuhiro A, Hiromoto R, Tu L. Branch-and-Bound Algorithm Applied to Uncertainty Quantification of a Boiling Water Reactor Station Blackout. Nuclear Engineering and Design 2015;295:283–304.
[30] Nielsen J, Tokuhiro A, Hiromoto R, Khatry J. Optimization Method to Branch-And-Bound Large SBO state Spaces under Dynamic Probabilistic Risk Assessment Via Use of LENDIT Scales and S2R2 Sets. Journal of Nuclear Science and Technology 2014;51(10):1212–30.
[31] Ladyman J, James L, Karoline W. What Is A Complex System? European Journal for Philosophy of Science 2013;3(nr. 1):33–67.
[32] Dynamic Positioning Basic Elements, https://www.youtube.com/watch?v=_RD_dehNnp0, 2020. [Internett].
[33] Coyne K. A Predictive Model of Nuclear Power Plant Crew Decision-Making and Performance in A Dynamic Simulation Environment. 2009.
[34] Lee JH, Yilmaz A, Denning R, Aldemir T. Use of Dynamic Event Trees and Deep Learning for Real-Time Emergency Planning in Power Plant Operation. Nuclear technology 2018.
[35] Loucks DP, Beek Ev. An Introduction to Optimization Models and Methods. Water Resource Systems Planning and Management. 2017. p. 93–177.
[36] Chang YHJ, Mosleh A. Cognitive Modeling and Dynamic Probabilistic Simulation of Operating Crew Response to Complex System Accidents: Part 3: IDAC Operator Response Model. Reliability Engineering & System Safety 2007;92(8):1041–60.
[37] Lee J, Leyffer S. Mixed Integer Nonlinear Programming, 154. Springer Science & Business Media; 2011.
[38] Gill PE, Wong E. Sequential Quadratic Programming Methods. i *Mixed Integer Nonlinear Programming*. New York, NY: Springer; 2012. p. 147–224.
[39] Exler O, Schittkowski K. A Trust Region SQP Algorithm for Mixed-Integer Nonlinear Programming. Optimization Letters 2007;1(nr. 3):269–80.
[40] Mitra G. Investigation of Some Branch and Bound Strategies for The Solution of Mixed Integer Linear Programs. Mathematical Programming 1973;4(nr. 1):155–70.
[41] Parhizkar T, Mosleh A, Roshandel R. Aging Based Optimal Scheduling Framework for Power Plants using Equivalent Operating Hour Approach. Applied Energy 2017;205:1345–63.
[42] Borchers B, Mitchell JE. An Improved Branch Aand Bound Algorithm for Mixed Integer Nonlinear Programs. Computers & Operations Research 1994;21(nr. 4):359–67.
[43] Eberhart R, Kennedy J. Particle Swarm Optimization. In: Proceedings of the IEEE international conference on neural networks; 1995.
[44] dos Santos Coelho L. An Efficient Particle Swarm Approach for Mixed-Integer Programming in Reliability–Redundancy Optimization Applications. Reliability Engineering & System Safety 2009;94(4):830–7.
[45] Parhizkar T, Utne I, Vinnem JE, Mosleh A. Supervised Dynamic Probabilistic Risk Assessment of Complex Systems, Part 2: Application to Risk-Informed Decision Making, Practice and Results. RESS Journal 2020;208(107392):1–16. https://doi.org/10.1016/j.ress.2020.107392. Submitted.
[46] Parhizkar T, Hogenboom S, Vinnem JE, Utne IB. Data Driven Approach to Risk Management and Decision Support for Dynamic Positioning Systems. Reliability Engineering & System Safety 2020;201:106964.
[47] Parhizkar T, Vinnem JE, Utne IB. Dynamic Probabilistic Safety Assessment Framework to Assist Decision-making in Complex Systems, Case Study: Dynamic Positioning Drilling Unit. Risk Analysis 2019. Submitted.
[48] Salciccioli JD, Crutain Y, Komorowski M, Marshall DC. Sensitivity Analysis and Model Validation. Secondary Analysis of Electronic Health Records. Springer; 2016. p. 263–71.
[49] Zio E. The Future of Risk Assessment. Reliability Engineering & System Safety 2018;177:176–90.
[50] Chang YHJ, Mosleh A. Cognitive Modeling and Dynamic Probabilistic Simulation of Operating Crew Response to Complex System Accidents: Part 1: Overview of The IDAC Model. Reliability Engineering & System Safety 2007;92(8):997–1013.